BAB 2
LANDASAN TEORI

2.1 Diabetes Mellitus

Diabetes mellitus (DM) adalah sebuah penyakit yang disebabkan oleh banyak faktor bisa dari kurangnya insulin atau ketidakmampuan tubuh untuk memanfaatkan insulin dengan simtoma berupa hiperglikemia kronis dan gangguan metabolisme karbohidrat, lemak dan protein sebagai akibat dari defisiensi atau aktivitas insulin.

Dalam melakukan diagnosis diabetes selain dari hasil yang didapatkan dari tes lab ada juga beberapa hal yang diperhatikan, seperti tekanan darah, glukosa, insulin, berat badan, keturunan, dan ketebalan kulit. Poin – poin tersebut dijadikan pertimbangan dalam melakukan diagnosis pasien.

Diabetes bisa dicegah dengan mengganti pola makan sampai pola hidup seseorang. Banyak teknik yang telah diterapkan untuk mengurangi resiko diabetes. Dasarnya pencegahan lebih dipilih, tapi untuk sekarang metode pengobatan saat ini masih belum sepenuhnya memadai [1].

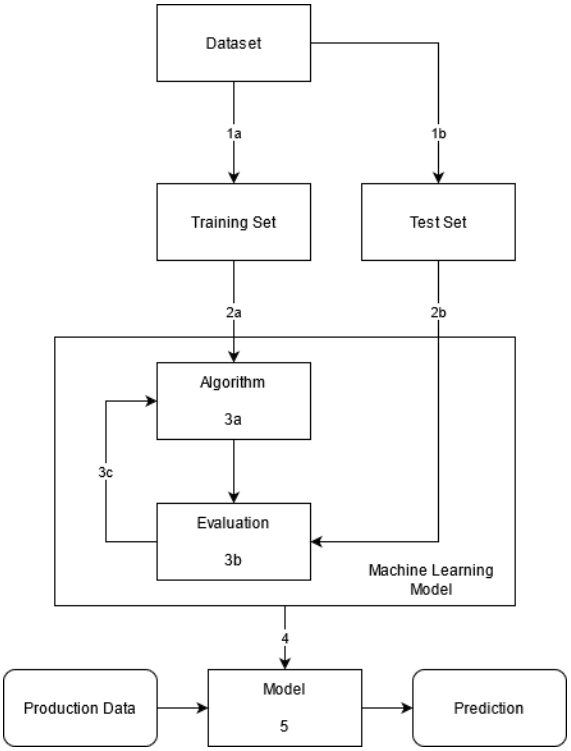
2.2 Machine Learning

Machine learning adalah sistem kecerdasan buatan (*artificial intelligence)* yang bisa "belajar" dari contoh – contoh yang dibuat bisa melakukan adaptasi atau "belajar" [4].

Machine learning memiliki tujuan untuk membuat suatu klasifikasi yang mudah dimengerti oleh manusia. *Machine learning* juga harus bisa meniru cara kerja nalar manusia agar bisa menjelaskan cara kerja pengambilan keputusan dalam melakukan pemecahan masalah [5].

Machine learning bekerja dengan cara "belajar" data pembelajaran atau bisa disebut *dataset* latih. Dengan melakukan pembelajaran, sistem yang dibuat menghasilkan model yang diharapkan kedepannya bisa digunakan untuk melakukan prediksi terhadap data – data baru yang mirip dengan *dataset* latih

yang diberikan. Setelah selesai melakukan pembelajaran, model yang dihasilkan dapat digunakan untuk mencari solusi data – data baru. Jika hasil dari prediksi kurang memadai, maka proses pelatihan model bisa dilakukan lagi sampai hasil yang didapat sesuai.



Gambar 2-1 Proses Machine Learning

Dengan Gambar 2-1 proses dari machine learning dapat dijelaskan sebagai berikut [4].

1. Dataset yang akan diteliti dibagi menjadi dua dataset baru, training dataset untuk pelatihan, dan test dataset untuk pengujian. (1a, 1b).
2. Dataset tersebut dipreprocessing untuk menghilangkan data yang kurang relevant (2a, 2b).
3. Dataset diolah dengan metode – metode machine learning yang dipilih (3a), dan kemudian dievaluasi hasilnya (3b). Jika tidak sesuai, data diproses kembali (3c).

4. Ketika metode dan hasil yang didapatkan sesuai, machine learning kemudian menghasilkan metode yang bisa digunakan untuk prediksi atau klasifikasi data yang mirip. (4).

Model yang dihasilkan digunakan untuk memprediksi atau klasifikasi data yang akan diprediksi atau klasifikasi (5).

2.3 Preprocessing

Preprocessing merupakan langkah yang dilakukan dalam metode machine learning dengan menghilangkan noise atau data – data yang kurang relevant. Proses ini dapat mempengaruhi performa machine learning itu sendiri dikarenakan machine learning hanya belajar dari data – data yang relevan.

Preprocessing memiliki contoh ketika data yang bersumber dari dunia nyata memiliki terlalu banyak fitur, tetapi dari fitur- fitur tersebut hanya beberapa yang berhubungan dengan hasil yang ingin dicari, hal ini menimbulkan redundancy dimana fitur-fitur yang kurang relevant tersebut akan digunakan untuk modeling dalam machine learning dan bisa mempengaruhi akurasi dan performansi metode machine learning yang digunakan. Dengan menggunakan preprocessing, fitur – fitur atau data yang kurang relevant bisa diminimalisasi atau dihilangkan yang membuat model machine learning yang digunakan lebih cepat dan akurat [6].

2.4 Feature Selection

Feature selection digunakan untuk memilih sebagian atau subset dari fitur atau variabel yang digunakan dalam suatu dataset. Proses ini dilakukan untuk mengurangi noise atau data dari variabel atau fitur yang kurang relevan untuk mendapatkan hasil atau performa yang lebih baik.

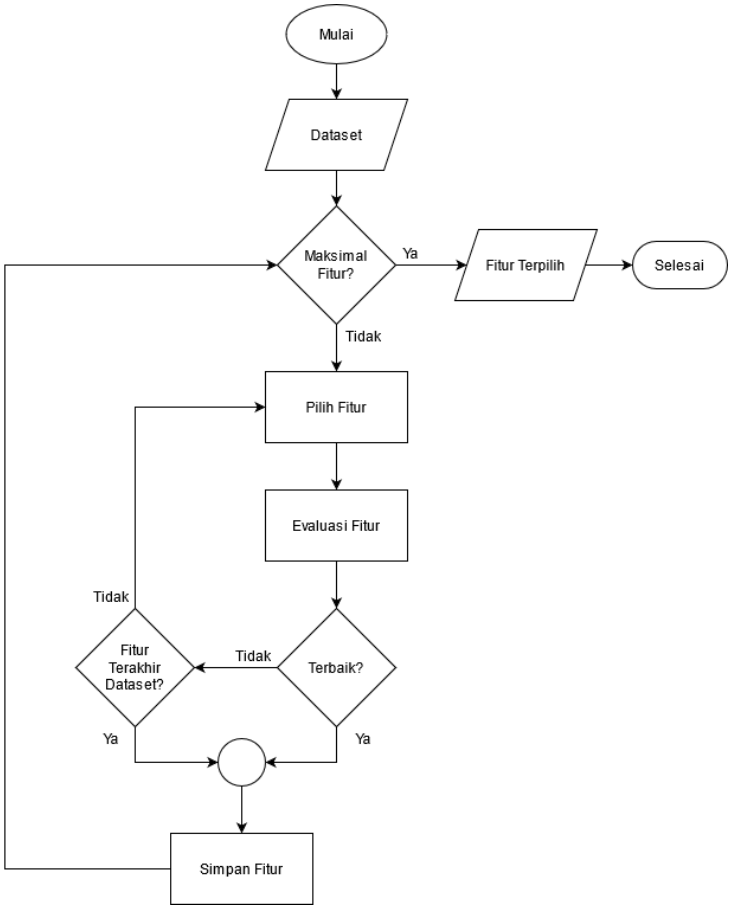
Salah satu contoh dari feature selection adalah di penelitian mikrogenetik. Dalam genetik, data bisa memiliki ratusan atau ribuan data dimana data – data tersebut bisa memiliki hubungan satu dengan yang lainnya. Tetapi data atau variabel – variabel tersebut bisa memiliki informasi lebih yang tidak berguna terhadap kelas dari data tersebut. Maka dari itu dengan menghilangkan variabel

tersebut jumlah data dan variabel yang digunakan akan berkurang dan performa dari klasifikasi data akan meningkat karena jumlah data yang berkurang [7]

2.5 Sequential Feature Selector

Sequential feature selection (SFS) adalah algoritma yang melakukan pencarian dari bawah ke-atas, yang bertujuan untuk mencari kombinasi fitur yang optimal dengan menggunakan fitur yang sesedikit mungkin.

Sequential feature selection bekerja dengan cara menguji dan mengkombinasi fitur yang ada di dataset satu – persatu dengan menggunakan suatu model klasifikasi [8].



Gambar 2-2 Alur Proses SFS

Dari gambar 2-2 dapat dijelaskan alur dari sequential forward selection sebagai berikut.

1. Dataset dibaca.
2. Melakukan pengecekan apakah fitur sudah sesuai dengan keinginan.

3. Jika tidak pemilihan fitur dimulai.
4. Fitur yang dipilih dievaluasi menggunakan suatu metode seperti metode klasifikasi.
5. Melakukan pengecekan apakah hasil evaluasi dari fitur yang dipilih merupakan fitur terbaik.
6. Jika tidak melakukan pengecekan apakah merupakan fitur terakhir dari dataset yang digunakan. Jika iya fitur yang dipilih akan disimpan.
7. Kemudian melakukan pengecekan kembali apakah jumlah fitur yang terpilih sesuai dengan keinginan, jika iya fitur terpilih akan disimpan dan bisa digunakan.

2.6 Normalisasi

Normalisasi adalah salah satu metode preprocessing data yang mengubah data awal ke dalam bentuk lain dengan tujuan untuk mendapatkan data yang lebih tepat untuk dianalisis dan pemodelan [9].

Proses normalisasi dilakukan dengan berfokus terhadap penskalaan data. Normalisasi data dapat membantu mencegah rentang data awalan yang besar agar tidak melebihi data dengan rentang awal yang lebih kecil dengan memberikan semua data bobot yang sama.

Berikut ini adalah beberapa cara yang dapat dilakukan untuk normalisasi data [9].

1. *Min-max normalization*

   Metode ini dilakukan dengan cara memproyeksi rentang data awal ke rentang data yang baru. Rentang data baru yang didapat umumnya adalah 0, 1] atau [-1, 1]. Metode normalisasi ini dilakukan dengan melakukan persamaan berikut.

$$v' = \frac{v - min_x}{max_x - min_x} (new\_max_x - new\_min_x) + new\_min_x \qquad (1)$$

Dimana:

v' Nilai baru hasil normalisasi.

v Nilai sebelum normalisasi.

min_x Nilai terkecil terhadap atribut x.

max_x Nilai terbesar terhadap atribut x.

new_min_x Nilai terkecil baru terhadap atribut min_x.

new_max_x Nilai terbesar baru terhadap atribut max_x.

Metode ini memiliki keuntungan dalam menjaga hubungan antara nilai data asli, tetapi jika inputan untuk normalisasi turun diluar rentang metode ini dapat menyebabkan kesalahan *out of bound*.

2. *Standardization*

Atau juga disebut *Z-Score normalization* merupakan metode normalisasi yang mengubah penyebaran data dan bersarnya data. Dalam metode ini, nilai yang akan diubah dinormalisasikan berdasarkan mean dan standar deviasi. Metode ini bisa dilakukan dengan menggunakan persamaan berikut

$$v' = \frac{v - \bar{x}}{\sigma_x} \quad (2)$$

Dimana:

v' Nilai baru hasil normalisasi.

v Nilai sebelum normalisasi.

$\bar{x}$ Mean dari x.

σ_x Standar deviasi dari x.

Untuk mendapatkan nilai mean, bisa dilakukan perhitungan dengan menggunakan persamaan berikut

$$\bar{x} = \frac{\sum_{i=0}^{n} x_i}{n} \quad (3)$$

Dimana:

$\bar{x}$ Nilai mean.

x_i Nilai x ke i.

n Jumlah data.

Untuk mendapat nilai standar deviasi dapat dilakukan perhitungan dengan menggunakan persamaan berikut

$$\sigma_x = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \quad (4)$$

Dimana:

σ_x Nilai standar deviasi.

x_i Nilai x ke i.

μ Mean dari keseluruhan.

N Jumlah data.

Jika perhitungan standar deviasi tidak dilakukan terhadap keseluruhan data atau data yang digunakan adalah sampel, perhitungan yang dilakukan dapat menggunakan persamaan berikut.

$$s_x = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n - 1}} \quad (5)$$

Dimana:

s_x Nilai standar deviasi.

x_i Nilai x ke i.

$\bar{x}$ Mean dari data sampel.

n Jumlah data.

Metode ini berguna ketika nilai minimal dan maksimal dari atribut *x* tidak diketahui, atau ketika ada *outliner* dalam metode *min-max normalization*.

3. *Decimal Scaling*

Metode ini dilakukan dengan cara memindahkan titik desimal dari suatu nilai atribut. Jumlah dari titik desimal yang dipindahkan bergantung terhadap nilai absolut maksimal atribut tersebut. Metode ini dapat dilakukan dengan menggunakan persamaan berikut

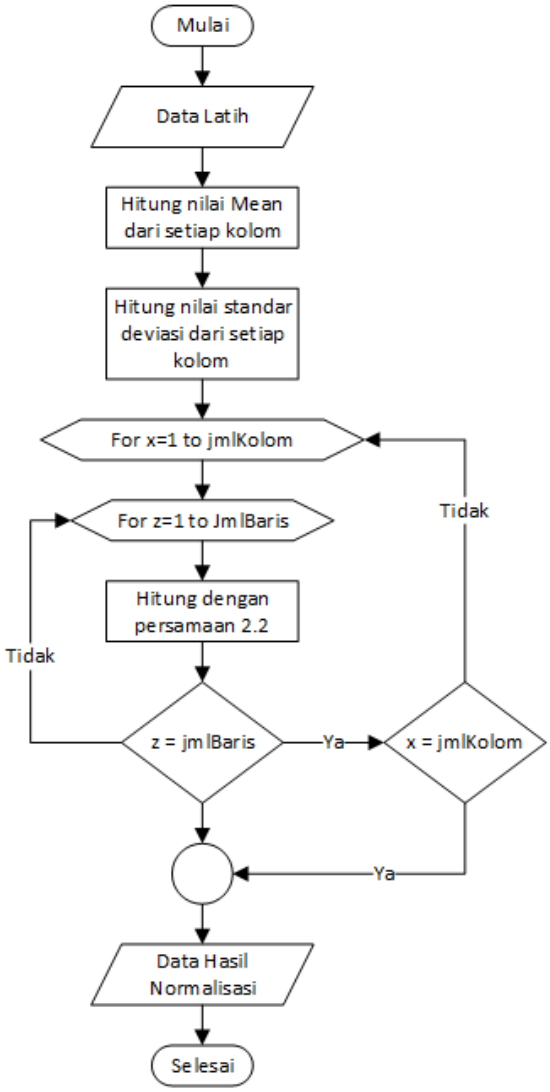
$$v' = \frac{v}{10^j} \quad (6)$$

Dimana:

v' Nilai baru normalisasi

v Nilai sebelum normalisasi

j Bilangan bulat terkecil $\max(|v'|) < 1$

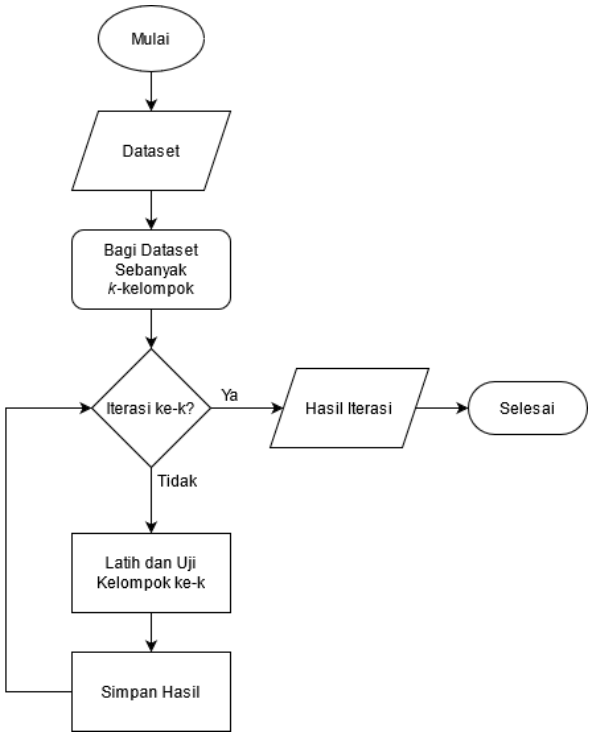


Gambar 2-3 Alur Proses Normalisasi

2.7 K-Fold Cross Validation

K-fold cross validation adalah suatu metode yang digunakan untuk mengevaluasi performa metode *machine learning*. Metode ini bekerja dengan cara menguji besarnya error pada data testing. Dalam cross validation, data dibagi sebanyak *k* bagian dengan ukuran sampel yang sama. Dari data sampel

tersebut kemudian dibagi lagi dengan ukuran *k-1* digunakan untuk pelatihan, sedangkan sisanya digunakan untuk pengujian. Pengujian di metode ini dilakukan sebanyak *k*-kali dengan mengganti – ganti partisi data yang digunakan [10].



Gambar 2-4 Alur Proses K-Fold Cross-Validation

2.8 Klasifikasi

Klasifikasi adalah "penyusunan bersistem dalam kelompok atau golongan menurut kaidah atau standar yang ditetapkan." Dalam *machine learning*, klasifikasi digunakan untuk mengkategorikan data menjadi *class – class* yang telah didefinisikan. Metode - metode yang digunakan klasifikasi seperti *pattern recognition* atau diskriminasi adalah bagian dari *supervised learning* [5].

Berikut ini adalah contoh dari bagaimana cara klasifikasi bekerja: sebuah dataset *email* yang terdiri dari alamat *email*, judul, konten dan *attachment* dengan *class spam* atau *non-spam* diproses menggunakan metode *pattern recognition* untuk melihat *email – email* mana yang merupakan *spam* atau *non-spam*, kemudian klasifikasi akan mengelompokan *email – email* mana saja yang

merupakan *spam* dan *non-spam* dari hasil metode *pattern recognition* tersebut. [4]

2.9 Supervised Learning

Supervised learning adalah sebuah metode *machine learning* yang bekerja menggunakan data atau model berlabel untuk melakukan klasifikasi. [7][8] Dalam ilmu statistika *supervised learning* biasanya dikenal dengan diskriminasi data, yang berarti melakukan klasifikasi menurut data yang telah diklasifikasikan dengan baik.

Supervised learning memiliki dua kategori, yaitu: *classification* dan *regression*

Cara kerja *supervised learning* adalah dengan menggunakan suatu *dataset* latih yang mengandung *label – label input* dan *output* (*class*). Hasil dari proses tersebut adalah sebuah model yang bisa digunakan untuk melakukan prediksi berdasarkan *dataset* baru. Dengan menggunakan *dataset* yang berjumlah besar akurasi dari model yang didapat bisa ditingkatkan [13]

2.10 Support Vector Machine

Support Vector Machine (SVM) adalah sebuah algoritma yang dikembangkan oleh Boser, Guyon, dan Vapnik di *Annual Workshop on Computational Learning Theory* pada tahun 1992. SVM menggabungkan 3 jenis idea: solusi dari *hyperplane* yang optimal, konvolusi dari *dot-product* (memperluas permukaan solusi linear dan non-linear), dan gagasan *soft-margin* (memungkinkan kesalahan pada *dataset* latih). Dilihat dari kesederhanaan SVM, algoritma ini menunjukan kinerja yang sangat baik. [14]

Cara kerja *SVM* dapat dijelaskan sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah antar *class* pada *input plane*. *Hyperplane* pemisah terbaik dapat ditemukan dengan mengukur *margin* dari *hyperplane* tersebut dan mencari titik maksimalnya. *Margin* merupakan jarak antar *hyperplane* dengan *pattern* terdekat dari masing – masing *class*. Perbedaan dari *SVM* dengan *neural network* lainnya adalah dalam pencarian *hyperplane* antara

class, dengan *SVM* lebih ke pencarian *hyperplane* yang terbaik dalam bagian *input space* [15]

Dalam *SVM* data yang tersedia dinotasikan sebagai $\vec{x_i} \in R^d$ dan label data sebagai $y_i \in$ {-1+1} untuk *i = 1, 2, ..., n* yang dimana *n* adalah banyak data. Diasumsikan jika kedua *class* yang ada -1 dan +1 dapat terpisah secara sempurna oleh *hyperplane* berdimensi *d*, dapat didefinisikan sebagai:

$$\vec{w} * \vec{x} + b = 0 \tag{7}$$

Pattern $\vec{x}$ yang merupakan *class* -1 dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan:

$$\vec{w} * \vec{x} + b \leq -1 \tag{8}$$

Dengan *pattern* $\vec{x}$ yang merupakan class +1 dapat dirumuskan sebagai berikut:

$$\vec{w} * \vec{x} + b \leq +1 \tag{9}$$

Margin terbesar dapat ditemukan dengan memaksimalkan jarak antar *hyperplane* dan titik terdekatnya. Hal ini termasuk dalam masalah *Quadratic Programming* yang merupakan pencarian titik minimal dengan memperhatikan constraint persamaan.

$$\min_{\vec{w}} \tau(w) = \frac{1}{2} \|\vec{w}\|^2 \tag{10}$$

$$y_i(\vec{x}_i . \vec{w} + b) - 1 \geq 0, \forall i \tag{11}$$

Ini dapat diselesaikan dengan teknik *Lagrange Multiplier*.

$$L(\vec{w}, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^{l} \alpha_i (y_i((\vec{x}_i, \vec{w} + b) - 1)) \tag{12}$$

$$(i = 1, 2, \dots, l)$$

Dimana α_i adalah *Lagrange multiplier* yang bernilai nol atau positif $(\alpha_i \geq 0)$. Nilai optimal bisa didapatkan dengan meminimalkan *L* terhadap $\vec{w}$ dan *b*, dan memaksimalkan *L* terhadap α_i

$$\sum_{i=1}^{l} a_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \vec{x}_i \vec{x}_j \tag{13}$$

Menjadi.

$$\alpha_i \geq 0 \ (i = 1, 2, \dots, l) \qquad \sum_{i=1}^{l} a_i y_i = 0 \tag{14}$$

Hasil dari perhitungan ini dapat memperoleh α_i yang bernilai positif. Data yang berkorelasi dengan α_i inilah yang disebut sebagai *support vector*. [16]

Singkatnya SVM adalah sebuah algoritma untuk memaksimalkan fungsi matematika yang menghormati data – data yang diberikan.

2.11 Smooth Support Vector Machine

Smooth Support Vector Machine (SSVM) adalah algoritma yang digunakan untuk melakukan klasifikasi atau regresi yang dikembangan dari metode *Support Vector Machine* (SVM) oleh Yuh Jye-Lee (2001), SSVM menggunakan metode pemisah *hyperplane* dalam melakukan klasifikasinya seperti SVM, namun SSVM menggunakan metode *smoothing* yang menggantikan *plus function* SVM dengan *integral* fungsi *sigmoid neural network*, metode ini digunakan untuk membuat SSVM efisien dalam *data* berdimensi tinggi. [10] Jika dibandingkan dengan metode yang lain SSVM memiliki keunggulan dalam kecepatan dan kemampuan generalisasi yang tinggi. [3]

Berikut ini adalah contoh permasalahan klasifikasi *m* pada ruang dimensi *n* real dengan notasi R^n, digambarkan dengan matriks *A* dengan dimensi *m x n* dengan kelas 1 atau -1 yang dispesifikasikan dengan matriks *D* diagonal *m x m*.

$$\min(\omega, \gamma, y) \frac{v}{2} y'y + \frac{1}{2}(\omega'\omega + \gamma^2) \tag{15}$$
$$D(A\omega - e\gamma) + y \geq e; y \geq 0$$

- w^b Seukuran inputan
- b^0 Angka
- *D* Matriks diagonal berukuran (m x m) dengan nilai [1, -1]
- *A* Matriks berukuran (m x n)
- *m* Banyak data
- *n* Fitur

W	Vector normal berukuran (n x 1)
e	Vektor berukuran (m x 1)
y	Parameter penentu lokasi bidang pemisah
v	Parameter positif penyeimbang bobot training error dan margin maximation term

Persamaan di atas membandingkan setiap elemen vektor. Jika kedua kelas dapat terpisah dengan sempurna oleh *hyperplane* yang terdefinisi dengan $x^T w + \gamma = 0$, maka terdapat dua bidang batas kedua kelas yang sejajar $x^T w + \gamma = +1$ untuk +1, dan $x^T w + \gamma = -1$ untuk -1.
Dengan solusi

$$y = \big(e - D(A\omega - e\gamma)\big) +, \tag{16}$$

Dengan mengubah komponen negatif dengan nol, bisa dilakukan perubahan y di (2) dengan $\big(e - D(Aw - ey)\big) +$ dan mengubah (1) menjadi SVM optimasi tanpa kendala.

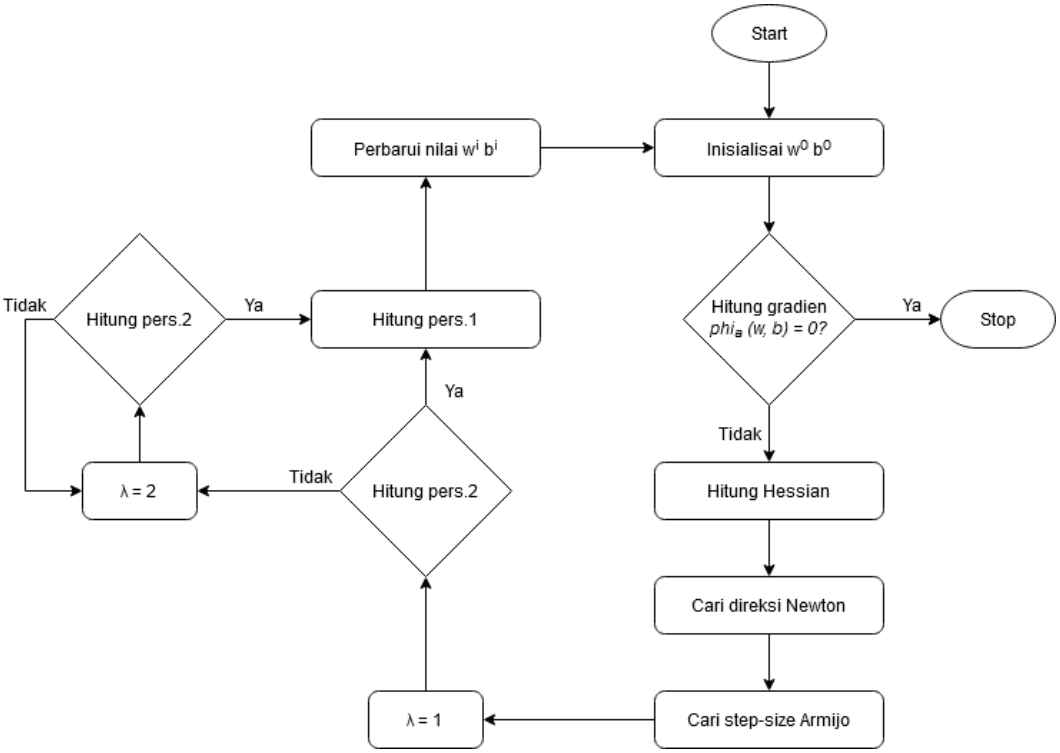
$$\min(\omega, y)\frac{v}{2}||\big(e - D(A\omega - e\gamma)\big) + ||_2^2 + \frac{1}{2}(\omega'\omega + \gamma^2) \tag{17}$$

Hasil ini memperlihatkan *convex minimization problem* tanpa konstrain, dengan ini dilakukannya proses smoothing yang menggantikan fungsi x_+ dengan fungsi *sigmoid neural network* $\frac{1}{1+e^{-ax}}$ dapat menghasilkan,

$$p(x, \alpha) = x + \frac{1}{\alpha}\log(1 + \varepsilon^{-\alpha x}), \alpha > 0 \tag{18}$$

Fungsi *p* dengan variabel α digunakan untuk menggantikan *plus function* di atas (2) agar mendapatkan fungsi SSVM sebagai berikut,
w^0 b^0 λ=1

$$\min_{(w,y)\in R^{n+1}} \phi_a(w, y) \coloneqq \min_{(w,y)\in R^{n+1}} \frac{v}{2}\|p(e - D(Aw - e\gamma), \alpha)\|_2^2 + \frac{1}{2}(w'w + \gamma^2)$$



Gambar 2-5 Diagram Algoritma SSVM

Dari grafik diatas (Gambar 2-5) proses dari klasifikasi dapat dilakukan dengan cara sebagai berikut.

1. Perhitungan Gradien

Perhitungan dilakukan dengan memformulasi *initial point* sebagai berikut.

w *matriks m x 1 disesuaikan dengan banyak fitur.*

y 0

Setelah melakukan formulasi initial point, perhitungan bisa dimulai menggunakan persamaan berikut.

$$\lim_{\alpha\to\infty} \nabla\Psi_\alpha\,(w,y) = \begin{bmatrix} w - vA^T D(e - D(Aw - ey)) \\ y + ve^T D(e - D(Aw - ey)) \end{bmatrix} \tag{19}$$

2. Pengecekan Gradien

Setelah melakukan perhitungan gradien, gradien kemudian dicek dengan menggunakan persamaan berikut.

$$\left\| \lim_{\alpha\to\infty} \nabla\Psi_\alpha\,(w,y) \right\|_2^2 \tag{20}$$

Jika proses ini terpenuhi, tahap dilanjutkan ke perhitungan matriks hessian.

3. Perhitungan Matriks Hessian

Perhitungan dilakukan dengan mencari komponen – komponen dari matriks hessian dengan menggunakan persamaan – persamaan berikut.

$$H_{1.1} = A^T\ diag\ (S_\infty(e - D(Aw - ey)))\ A \quad (21)$$

$$H_{1.2} = A^T\ diag\ (S_\infty(e - D(Aw - ey)))\ e \quad (22)$$

$$H_{2.1} = e^T\ diag\ (S_\infty(e - D(Aw - ey)))\ A \quad (23)$$

$$H_{2.2} = e^T\ diag\ (S_\infty(e - D(Aw - ey)))\ e \quad (24)$$

Setelah mendapatkan hasil – hasilnya, Langkah berikutnya adalah membuat matriks hessian itu sendiri dengan menggabungkan hasil dari $H_{1.1}$ sampai $H_{2.2}$ dengan menggunakan persamaan berikut.

$$\lim_{a\to\infty} \nabla^2\ \Psi_a(w,y) = \begin{bmatrix} \dfrac{\partial^2\Psi_a(w,y)}{\partial^2 w^2} & \dfrac{\partial^2\Psi_a(w,y)}{\partial w \partial y} \\ \dfrac{\partial^2\Psi_a(w,y)}{\partial^2 \partial w} & \dfrac{\partial^2\Psi_a(w,y)}{\partial^2 y^2} \end{bmatrix} = I + v \begin{bmatrix} H_{1.1} & H_{1.2} \\ H_{2.1} & H_{2.2} \end{bmatrix} \quad (25)$$

4. Menentukan Newton Direction

Proses ini dilakukan untuk menentukan step size dan initial point baru.

5. Klasifikasi

Setelah semua proses selesai, proses klasifikasi bisa dilakukan terhadap data yang dipilih.

2.12 Confusion Matrix

Confusion matrix atau error matrix digunakan untuk memvisualisasikan performa dari metode machine learning yang digunakan. Setiap baris dari matriks mewakili kelas dari dataset yang diuji, sedangkan kolom dari matriks mewakili kelas sebenarnya dari dataset. [17]

	+R	-R
+P	tp	fp
-P	fn	tn

Elemen – elemen dari confusion matrix dapat dijelaskan sebagai berikut.

+P Kelas dari prediksi data.

-P Kelas dari prediksi data.

+R	Kelas sebenarnya dari dataset.
-R	Kelas sebenarnya dari dataset.
tp	"True Positive" jumlah hasil prediksi yang benar.
fp	"False Positive" jumlah hasil prediksi benar yang salah.
fn	"False Negative" jumlah hasil prediksi yang salah.
tn	"True Negative" jumlah hasil prediksi salah yang benar.

Dari matriks atau tabel 2-1 dapat dilakukan perhitungan untuk mencari nilai performansi dengan menggunakan persamaan – persamaan berikut.

1. True Positive Rate

   Digunakan untuk mencari nilai probabilitas dari prediksi.

$$\frac{\sum tp}{\sum Nilai\ Positive} \tag{26}$$

2. False Positive Rate

   Digunakan untuk mencari nilai dari probabilitas *false alarm*.

$$\frac{\sum fp}{\sum Nilai\ Negative} \tag{27}$$

3. False Negative Rate

   Digunakan untuk mencari nilai *miss rate*.

$$\frac{\sum fn}{\sum Nilai\ Positive} \tag{28}$$

4. True Negative Rate

   Digunakan untuk mencari nilai *specificity* dan *selectivity*.

$$\frac{\sum tn}{\sum Nilai\ Negative} \tag{29}$$

5. Positive Predictive Value

   Digunakan untuk mencari nilai *precision.*

$$\frac{\sum tp}{\sum Prediksi\ Positive} \tag{30}$$

6. False Discovery Rate

   Digunakan untuk mencari nilai probabilitas dari kesalahan.

$$\frac{\sum fp}{\sum Prediksi\ Positive} \tag{31}$$

7. False Omission Rate

   Digunakan untuk mencari nilai proporsi dari hasil negatif dan positif.

$$\frac{\sum fn}{\sum Prediksi\ Negative} \tag{32}$$

8. Negative Predictive Value

   Digunakan untuk mencari nilai *Negative predictive value.*

$$\frac{\sum tn}{\sum Prediksi\ Negative} \tag{33}$$

2.13 Penelitian Terkait

Terdapat beberapa penelitian yang sudah dilakukan sebelumnya dengan topik yang sama tetapi menggunakan algoritma – algoritma yang berbeda, berikut ini adalah beberapa hasil rangkuman dari studi literatur yang telah dilakukan..

2.13.1 A New Smooth Support Vector Machine and Its Applications in Diabetes Disease Diagnosis

Penelitian ini menggunakan algoritma *Multiple Knot Spline SSVM* (MKS-SSVM). yang merupakan pengembangan dari algoritma SSVM, dengan dataset yang digunakan adalah PID (Pima Indian Diabetes Dataset).

MKS-SSVM ini menghasilkan akurasi 93.2% dengan catatan *"Hasil dari penelitian ini menunjukan MKS-SSVM menghasilkan hasil yang efektif dalam mendeteksi diagnosa diabetes yang dibandingkan dari penelitian sebelumnya.*

Namun waktu komputasi SSVM masih lebih unggul dibandingkan dengan MKS-SSVM". [2]

Tabel 2-1 Hasil Perbandingan dari SSVM dan MKS-SSVM

	SSVM	**MKS-SSVM**
Best parameter (C,γ)	(1.78, 3.37e-005)	(316.23, 0.14)
Training Accuracy (%)	77.66	93.24
Testing Accuracy (%)	76.73	93.20
CPU Time (sec)	399.762	700.1793

2.13.2 Klasifikasi Pasien Diabetes Mellitus Menggunakan Metode Smooth *Support Vector Machine* (SSVM)

Penelitian ini menggunakan algoritma *SSVM* dengan *kernel Gaussian Radial Basis Function* (RBF).

Dataset yang digunakan adalah dataset sekunder dari pegawai Kementerian Perindustrian di Balai Kesehatan Kementerian Perindustrian dari Juli 2014 sampai September 2014 dengan jumlah data sebanyak 496 data.

Hasil dari penelitian ini menunjukan penggunaan *SSVM* dengan *kernel Gaussian Radial Basis Function* (RBF) yang memberikan hasil akurasi sebesar **97.03%**, dengan hasil kelas positif dan negatif sebesar **98,33%** dan **95,12%.** [10]

2.13.3 Application of Genetic Algorithm Optimized Neural Network Connection Weights for Medical Diagnosis of Pima Indians Diabetes

Penelitian yang dilakukan oleh Asha Gowda Karegowda, A.S. Manjunath, dan M.A. Jayaram ini melakukan penelitian menggunakan *dataset* PID dengan menggunakan algoritma *Back Propagation Network*, dan *hybrid Genetic Algorithm* dengan *Back Propagation Network* (GA-BPN). [16]

Tabel 2-2 Hasil Perbandingan Back Propagation Network dan Hybrid Genetic Algorithm dan Back Propagation Network

Attribute Selection Method	Number of Attribute	BPN Accuracy (%)	GA-BPN Accuracy (%)
With All Attributes	8	72.88	77.707
Decision Tree	5	78.21	84.076
GA-CFS	4	79.5	84.713

2.13.4 Comparison of Classifiers for the Risk of Diabetes Prediction

Penelitian yang dilakukan oleh Nongyao Nai-aruna, Rungruttikarn Moungmaia ini melakukan perbandingan antara beberapa algoritma klasifikasi seperti: *Decision Tree, Artificial Neural Network, Logistic Regression, Naïve Bayes*, dan *Random Forest*. Untuk *dataset* yang digunakan untuk penelitian ini adalah dari 26 *Primary Care Units* (PCU) di *Sawanpracharak Regional Hospital* pada periode 2012 – 2013 dengan variabel - variabel di tabel 2-4. [18]

Tabel 2-3 Variabel - Variabel dan Class Dataset yang Digunakan

Variable	Keterangan	Value
BMI	Body Mass Index (kg/m^2)	numeric
AGE	Age (year)	numeric
WAIST_CM	Height (cm)	numeric
BPH	Systolic blood pressure (mmHg)	numeric
BPL	Diastolic blood pressure (mmHg)	numeric
DM_FAMILY	History of Diabetes Family	1: Have 2: No Have 9: Unknown
HT_FAMILY	History of Hypertension Family	1: Have 2: No Have 9: Unknown
ALCOHOL	Alcohol drinking	1: No Alcohol 2: No Have

		3: Occasionally 4: Often 9: Unknown
SMOKING	Smoking behaviour	1: No Smoke 2: No Have 3: Occasionally 4: Often 9: Unknown
Sex	Sex	1: Female 2: Male
CLASS	**1: Normal Group** **2: Diabetes risk group**	

Dataset yang digunakan mempunyai jumlah data sebanyak 30.122 orang yang dibagi menjadi dua grup yaitu: 19.145 orang di grup normal, dan 10.977 orang di grup dengan resiko diabetes.

Berikut ini adalah hasil klasifikasi yang telah dilakukan dengan menggunakan 13 jenis algoritma dengan *Random Forest* yang memiliki hasil terbaik sebesar 85.558 %.

Tabel 2-4 Hasil Perbandingan 13 Algoritma dengan Dataset 26 Primary Care Units (PCU) di Sawanpracharak Regional Hospital

Model	Accuracy (%)
Decision Tree	85,090
Artificial Neural Network	84,532
Logistic Regression	82,308
Naïve Bayes	81,010
Bagging dengan Decision Tree (BG-DT)	85,333
Bagging dengan Artificial Neural Network (BG-ANN)	85,324
Bagging dengan Logistic Regression (BG-LG)	82,318

Bagging dengan Naïve Bayes (BT-NB)	80,960
Boosting dengan Decision Tree (BT-DT)	84,098
Boosting dengan Artificial Neural Network (BT-ANN)	84,815
Boosting dengan Logistic Regression (BT-LG)	82,312
Boosting dengan Logistic Regression (BT-LG)	81,019
Random Forest	85,558

2.13.5 Diabetes Prediction Using Feature Selection and Classification

Penelitian oleh Khyati K. Gandhi, dan Prof. Nilesh B.Prajapati ini menggunakan metode *feature selection* (*F-score* dan *K-means clustering*) sebelum dinormalisasi menggunakan *Z-score* dan kemudian diproses dengan *SVM*. Penelitian ini adalah *dataset* PID. [19]

Tabel 2-5 Hasil Prediksi Menggunakan Feature Selection dan SVM

Dataset	Accuracy (%)	Sensitivity (%)	Specificity (%)
PID Dataset	98	97.77	97.79

2.13.6 Improved J48 Classification Algorithm for the Prediction of Diabetes

Penelitian oleh Gaganjot Kaur dan Amit Chhabra menggunakan algoritma J48 yang ditingkatkan dengan menggunakan API WEKA. Untuk *dataset* yang digunakan penelitian ini menggunakan Pima Indians Diabetes Dataset (PID).

Penelitian ini memberikan perbandingan antara J48 dan J48 dengan WEKA dengan hasil akurasi yang lebih tinggi yaitu J48 dengan WEKA. [20]

Tabel 2-6 Hasil Perbandingan J48 dengan J48+WEKA

Algorithm	Accuracy (%)	Error (%)
J48	73.8281	26.1719

J48 + WEKA	99.8700	0.1300

2.13.7 Medical Diagnosis on Pima Indian Diabetes Using General Regression Neural Network

Penelitian oleh Kamer Kayaer dan Tulay Yuldirim menggunakan *General Regression Neural Network* untuk prediksi pada *dataset Pima Indian Diabetes Dataset* (PID). Selain melakukan penelitian menggunakan *General Regression Neural Network*, pada penelitian ini juga ditampilkan hasil penelitian menggunakan *Radial Basis Function* (RBF) dan *Multilayer Perceptron* (MLP). [21]

Tabel 2-7 Hasil Penelitian Menggunakan RBF dan GRNN

Algorithm	Correct on Training (%)	Correct on Test (%)	Mean
RBF	100	68.23	92.06
GRNN	82.99	80.21	82.29

2.13.8 Performance Evaluation of Levenberg-Marquardt Technique in Error Reduction for Diabetes Condition Classification

Penelitian oleh Nawaz Khan, Dhara Guarav dan Thomas Kandl melakukan evaluasi teknik *Levenberg-Marquardt* (LM) yang difokuskan di algoritma *Multi-layer Perceptron* (MLP) untuk mencari algoritma yang errornya tidak melebihi 0.1 dengan *dataset* yang digunakan *Pima India Diabetes Dataset* (PID).

Penelitian dilakukan dengan membagi *dataset* menjadi tiga yaitu, yang pertama *dataset* untuk pelatihan, kedua untuk validasi silang agar tidak terjadi *overfiting*, dan set terakhir digunakan untuk melihat apakah algoritma sudah mempelajari pola latihan yang diberikan. Berikut ini adalah hasil dari penelitian tersebut. [22]

Tabel 2-8 Hasil Penelitian Kuatnya Algoritma

Dataset	Evaluation Matriks	LM	Delta-by-Delta	Quickprop	Momentum
Test	Sensitivity	58.18%	0%	0%	20%
	Specificity	90.90%	100%	100%	82.82%
Cross-Validated	Sensitivity	100%	58.33%	66.66%	58.33%
	Specificity	94.44%	27.77%	22.22%	22.22%

2.13.9 Predicting Diabetes Mellitus with Machine Learning Techniques

Penelitian oleh Quan Zou, Kalyang Qu, Yamel Luo, Dehul Yin dan Hua Tang ini melakukan penelitian menggunakan beberapa teknik *machine learning*, seperti: *Decision Tree* dengan metode *J48, Random Forest, dan Neural Network*.

Penelitian ini menggunakan dua *dataset*, *dataset primary* dari rumah sakit – rumah sakit di Luzhou, China yang terdiri dari 164431 yang akan digunakan sebagai *dataset* latih, dan 13700 digunakan sebagai *dataset* uji.

Proses penelitian dilakukan dengan memisahkan data yang digunakan dengan mengambil beberapa fitur atau menggunakan semua fitur yang tersedia oleh *dataset*, ini memungkinkan penelitian mendapat hasil yang bervariasi. Penelitian ini juga memproses data melalui *feature selection* untuk perbandingan. [23]

Tabel 2-9 Hasil Penelitian di Dataset Luzhou

Method	Algorithm	Accuracy
11 features	Neural Network	0.6983
	Decision Tree (J48)	0.6916
	Random Forest	0.7104
Without blood glucose	Neural Network	0.6986
	Decision Tree (J48)	0.6917
	Random Forest	0.7225

mRMR (minimum Redundancy Maximum Relevance)	Neural Network	0.757
	Decision Tree (J48)	0.7613
	Random Forest	0.7508
PCA (Principal Component Analysis)	Neural Network	0.7414
	Decision Tree (J48)	0.7388
	Random Forest	0.7395
With blood glucose	Neural Network	0.7572
	Decision Tree (J48)	0.761
	Random Forest	0.7597
All features	Neural Network	0.7841
	Decision Tree (J48)	0.7853
	Random Forest	0.8084

Tabel 2-10 Hasil Penelitian di Dataset PID

Method	**Algorithm**	**Accuracy**
mRMR (minimum Redundancy Maximum Relevance)	Neural Network	0.739
	Decision Tree (J48)	0.7534
	Random Forest	0.7721
PCA (Principal Component Analysis)	Neural Network	0.7475
	Decision Tree (J48)	0.7167
	Random Forest	0.7144
With blood glucose	Neural Network	0.7198
	Decision Tree (J48)	0.6895
	Random Forest	0.6728
All features	Neural Network	0.7667
	Decision Tree (J48)	0.7275
	Random Forest	0.7604

2.13.10Machine Learning and Data Mining Methods in Diabetes Research

Penelitian oleh Ioannis Kavakiotisa, Olga Tsavec, Athanasios Salifoglouc, Nicos Maglaveras, Ioannis Vlahavasa, dan Ioanna Chouvarda ini menjelaskan

bahwa beberapa algoritma dan metode bisa digunakan untuk melakukan penelitian *machine learning* dan data *mining* dengan topik diabetes.

Penelitian ini juga memberikan tabel perbandingan beberapa algoritma dengan contoh tipe kasus diabetesnya. [24]

Tabel 2-11 Hasil Perbandingan Beberapa Publikasi Penelitian Diabetes

Publication	Type 2 diabetes biomarkers of human gut micro-biota selected via iterative sure independent screening method.
Diabetes Type	T2D
Datasets	Gut microbiota on 2 datasets (A: 344, B: 145)
Algorithm Used	Logistic regression (LR), Linear Discriminant Analysis (LDA), Naïve Bayes (NB), Support Vector Machine (SVM).
Validation	10-fold Cross-validation
Best Accuracy	SVM
Publication	Non-invasive detection of fasting blood glucose level via electrochemical measurement of saliva.
Diabetes Type	Hyperglycaemia
Datasets	Electrochemical measurements of saliva (175 subjects)
Algorithm Used	Logistic regression (LR), Support Vector Machine (SVM), Artificial Neural Network (ANN)
Validation	3-fold Cross-validation
Best Accuracy	SVM (84.09%)

Publication	*Predictive models to assess risk of type 2 diabetes, hypertension and comorbidity: machine-learning algo-rhythms and validation using national health data from Kuwait—a cohort study.*
Diabetes Type	T2D
Datasets	Demographic, anthropometric, vital signs, diagnostic and clinical laboratory measurements (10632 subjects)
Algorithm Used	Logistic regression (LR), k-nearest neighbour (k-NN) ,multifactor dimensionality reduction (MDR) support vector machines (SVM)
Validation	5-fold Cross-validation
Best Accuracy	SVM (81.3%)
Publication	*Mani S, Chen Y, Elasy T, Clayton W, Denny J. Type 2 diabetes risk forecasting from EMR data using machine learning. AMIA Annu Symp Proc 2012; 2012:606–15 [Epub 2012 Nov 3].*
Diabetes Type	T2D
Datasets	Demographic, clinical lab values (2280 subjects distributed in three datasets)
Algorithm Used	Gaussian Naïve Bayes (NB), Logistic Regression (LR), K-nearest neighbour (k-NN, CART, Random Forests (RF), Support Vector Machine (SVM)
Validation	5-fold Cross-validation
Best Accuracy	RF (0.803/0.807/0.877)

Publication	*Tapak L, Mahjub H, Hamidi O, Poorolajal J. Real-data comparison of data mining methods in prediction of diabetes in Iran. Healthc Inform Res Sep 2013;19(3):177–85.* *http://dx.doi.org/10.4258/hir.2013.19.3.177 [Epub 2013 Sep 30].*
Diabetes Type	Nonspecified
Datasets	Demographic, anthropometric, diagnostic and clinical laboratory measurements (6500 subjects)
Algorithm Used	Artificial neural networks (ANN), support vectormachines (SVM), fuzzy c-mean, Random Forests (RF)
Validation	10-fold Cross-validation
Best Accuracy	SVM (0.986)

2.13.11Computational Intelligence in Early Diabetes Diagnosis: A Review

Penelitian oleh Shankaracharya, Devang Odedra, Subir Samanta, dan Ambarish S. Vidyarthi ini melakukan *review* terhadap penelitian – penelitian *machine learning* untuk topik diabetes yang sudah dilakukan dengan memaparkan *dataset* yang digunakan beserta algoritmanya. [1]

Tabel 2-12 Hasil Review

Algorithm	Dataset	Accuracy	Specificity	Sensitivity
MFNNCA	PID	80.07	83.38	74.00
GRG2	PID	81.25	-	-
ANFIS	PID	98.14	98.58	96.97
GRNN	PID	80.21	-	-
MLP	PID	77.08	-	-

RBF	PID	68.23	-	-
ARTMAP-IC	PID	81.00	-	-
MEA	PID	80.07	-	-
ESOM	PID	78.40	-	-
GNG	PID	74.60	-	-
GCS	PID	73.80	-	-
k-NN	PID	77.00	-	-
k-NN	PID	71.90	-	-
CART	PID	72.80	-	-
MLP	PID	75.20	-	-
LVQ	PID	75.80	-	-
LDA	PID	77.50	-	-
CART-DB	PID	74.40	-	-
SVM	Questionnaire	94.00	94.00	93.00
SSVM	PID	76.73	-	-
MKS-SSVM	PID	93.20	-	-
GDAL and S-SVM	PID	78.21	-	-
PCA-ANFIS	PID	89.47	-	-
LDA-ANFIS	PID	84.61	85.18	83.33
Naïve Bayes	PID	74.50	-	-
Semi-naïve Bayes	PID	76.00	-	-
C4.5	PID	76.00	-	-
MLPNN	PID	91.53	91.19	92.42
ME	PID	97.93	98.01	97.73
MME	PID	99.17	99.43	98.48

Penelitian ini juga memberikan kelebihan dan kekurangan untuk penelitian prediksi diabetes.

Tabel 2-13 Perbandingan Kelebihan dan Kekurangan

Algorithm	**Advantages**	**Disadvantages**
Back propagation	Better error minimization	Slow convergence rate
LM	Fast convergence rate	Memorization effect on overtraining

SVM	Guaranteed global minimum.	No specific rule to choose kernel that will give better classification.
ANFIS	Fast convergence rate	Low interpretability of learned in-formation, computationally expensive.
RBF	Uses small numbers of locally tuned units and is adaptive in nature	Sensitive to dimensionality of data.
ARTMAP-IC	Fast convergence rate	Tends to be conservative which reduces sensitivity
SOM	Little computational and memory requirements	Topology mismatch leads to poor classification
ESOM	Shorter learning process than SOM	Poor adaptability to input data
GNG	Can adaptively determine the number of connections	Poor response to changing inputs
k-NN	Good choice when there is no prior knowledge of data distribution	Requires rigorous tuning to optimally fit the real-world data
LVQ	Little computational and memory requirements	Less accurate with high dimensional data
LDA	Works best when class has Gaussian density	Less accurate with small sampel size
ME	Requires only small number of connections in neural network	Learns only static input-output mappings (i.e. no feedback)
MME	Requires only small number of connections in neural network. Faster than ME	Learns only static input-output mappings